Synopsis On

Heart Disease Prediction System

Submitted by
Shambhavi Gupta
Rajbala Bhadu
Shaista Parveen
Navya Srivastava

For the award of the degree

Of

B. Tech
(Computer Science)

Under Supervision of

Dr. Swati Nigam, Professor, Banasthali Vidyapith, Jaipur

1. INTRODUCTION

According to the World Health Organization, every year 12 million deaths occur worldwide due to heart disease. Heart disease is one of the biggest causes of morbidity and mortality among the population of world. Prediction of cardiovascular disease is regarded as one of the most important subjects in the section of data analysis. A load of cardiovascular diseases has rapidly increased all over the world in the past few years. Many types of research have been conducted to pinpoint the most influential factors of heart disease as well as accurately predict the overall risk. Heart Disease is even highlighted as a silent killer which leads to the death of the person without obvious symptoms. The early diagnosis of heart disease plays a vital role in making decisions on lifestyle changes in high-risk patients and in turn, reduces complications.

Heart disease is a major public health issue that affects millions of people worldwide. Early detection and prediction of the disease can greatly improve patient outcomes, which is why it is important to develop accurate and efficient methods to identify individuals at risk. Machine learning is a powerful tool that can be used to analyze large amounts of data and make predictions about future outcomes.

Machine learning proves to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry. This project aims to predict future heart disease by analyzing data of patients that classify whether they have heart disease or not using the machine-learning algorithm. Machine Learning techniques can be a boon in this regard.

By collecting the data from various sources, classifying them under suitable headings & finally analyzing them to extract the desired data we can say that this technique can be very well adapted to the prediction of heart disease.

1.1 PROBLEM STATEMENT

The major challenge in heart disease is its detection. There are instruments available that can predict heart disease but either they are expensive or are not efficient to calculate the chance of heart disease in humans. Early detection of cardiac diseases can decrease the mortality rate and overall complications.

However, it is not possible to monitor patients every day in all cases accurately and consultation with a patient for 24 hours by a doctor is not available since it requires more patience, time, and expertise. Since we have a good amount of data in today's world, we can use various machine-learning algorithms to analyze the data for hidden patterns. The hidden patterns can be used for health diagnosis in medicinal data.

2. LITERATURE REVIEW

With growing development in the field of medical science alongside machine learning various experiments and research has been carried out in recent years releasing the relevant significant papers.

[1] Santhana Krishnan. J, et al proposed a paper "Prediction of Heart Disease Using Machine Learning Algorithms' using a decision tree and Naive Bayes algorithm for the prediction of heart disease. In the decision tree algorithm, the tree is built using certain conditions which give True or False decisions. The algorithms like SVM, and KNN are results based on vertical or horizontal split conditions depending on dependent variables. But a decision tree for a tree-like structure having root nodes, leaves, and branches based on the decision made in each of tree Decision tree also helps in the understanding of the importance of the attributes in the dataset. They have also used the Cleveland data set. Dataset splits into 70% training and 30% testing by using some methods. This algorithm gives 91% accuracy. The second algorithm is Naive Bayes, which is used for classification. It can handle complicated, nonlinear, dependent data so it is found suitable for the heart disease dataset as this dataset is also complicated, dependent, and nonlinear in nature. This algorithm gives an 87% accuracy.

[2] Sonam Nikhar et al proposed a paper "Prediction of Heart Disease Using Machine Learning Algorithms" their research gives point to point explanation of Naïve Bayes and decision tree classifiers that are used especially in the prediction of Heart Disease. 3 Some analysis has been led to think about the execution of a prescient data mining strategy on the same dataset, and the result decided that Decision Tree has higher accuracy than the Bayesian classifier.

[3] Avinash Golande et al, proposed "Heart Disease Prediction Using Effective Machine Learning Techniques" in which a few data mining techniques are used that support the doctors to differentiate heart disease. Usually utilized methodologies are k-nearest neighbor, Decision Tree, and Naïve Bayes. Other unique characterization-based strategies utilized are packing calculation, Part thickness, consecutive negligible streamlining and neural systems, straight Kernel self-arranging guide, and SVM (Bolster Vector Machine).

[4] Lakshmana Rao et al, proposed "Machine Learning Techniques for Heart Disease Prediction" in which the contributing elements for heart disease are more. So, it is difficult to distinguish heart disease. To find the seriousness of heart disease among people different neural systems and data mining techniques are used.

3. MOTIVATION

The main motivation for doing this project is to present a heart disease prediction model for the prediction of the occurrence of heart disease. Further, this is aimed at identifying the best classification algorithm for identifying the possibility of heart disease in a patient. This work is justified by performing a comparative study and analysis using two classification algorithms namely Naïve Bayes, and SVM (Support Vector Machine).

The motivation for using machine learning in heart disease prediction is to improve the accuracy of diagnosis and to identify individuals at high risk of developing heart disease. This can help to prevent or delay the onset of heart disease by allowing for earlier intervention and treatment.

Heart disease is a leading cause of death worldwide, and early detection and diagnosis are crucial for reducing mortality rates. Traditional methods of diagnosis, such as physical examination and biomarker testing, can be limited in their accuracy and may miss early signs of heart disease. Machine learning, on the other hand, can analyze large amounts of data, including demographic information, medical history, and imaging results, to identify patterns and predict the likelihood of heart disease.

Overall, the use of machine learning in heart disease prediction can improve the early detection of heart disease, and help to reduce the impact of this leading cause of death worldwide.

4. OBJECTIVES

The main objective of a heart disease prediction system using machine learning is to accurately identify individuals who are at high risk of developing heart disease.

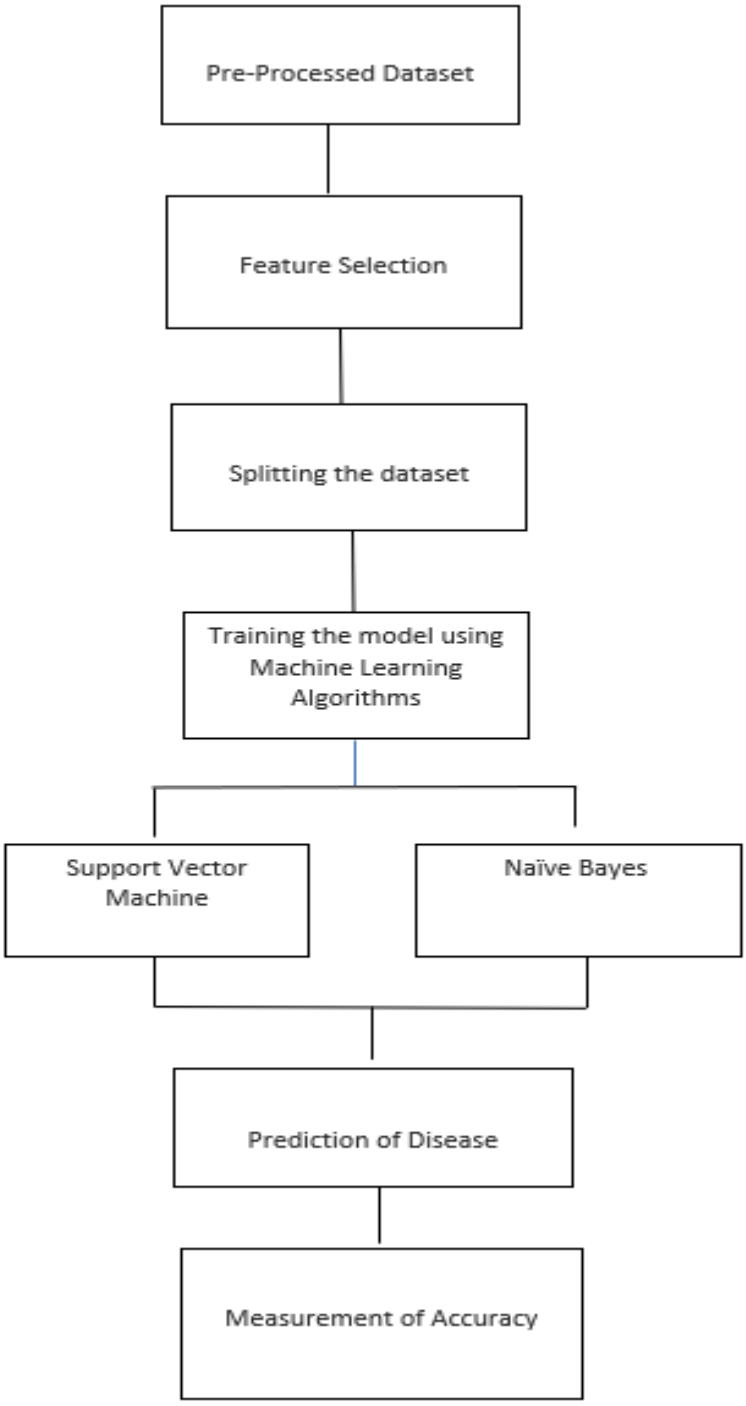
The goal is to enable earlier intervention and treatment for those at high risk, which can help to prevent or delay the onset of heart disease and ultimately, reduce the overall mortality rates from heart disease.

The objective of a heart disease prediction system using machine learning is to accurately predict the likelihood of an individual developing heart disease. The system aims to identify individuals at high risk of developing heart disease so that preventative measures can be taken and early treatment can be provided.

In summary, the objective of a heart disease prediction system using machine learning is to accurately identify high-risk individuals and provide early intervention and treatment to prevent or delay the onset of heart disease and ultimately, reduce the overall mortality rates from heart disease.

5. METHODOLOGY

5.1 Block Diagram



5.2 Data Collection

There are many databases related to heart diseases, such as the Cleveland database and the heart disease database provided by the National Cardiovascular Disease Surveillance System. This paper uses a widely used heart disease dataset from Kaggle, composed of four databases: Cleveland, Hungary, Switzerland, and the VA Long Beach. The dataset has 14 attributes, and each attribute is set with a value. It contains 1025 patient records of different ages, of which 713 are male, and 312 are female. This dataset is a subset of the original dataset containing 76 attributes, but most scholars only use 14 of them, since other attributes have little effect on heart diseases, such as time of exercise, ECG reading, and exercise protocol. The descriptions in this database are shown

Dataset Link

https://www.kaggle.com/datasets/puspitasaha/heart-disease-prediction

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	52	1	0	125	212	0	1	168	0	1.0	2	2	3	0
1	53	1	0	140	203	1	0	155	1	3.1	0	0	3	0
2	70	1	0	145	174	0	1	125	1	2.6	0	0	3	0
3	61	1	0	148	203	0	1	161	0	0.0	2	1	3	0
4	62	0	0	138	294	1	1	106	0	1.9	1	3	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1020	59	1	1	140	221	0	1	164	1	0.0	2	0	2	1
1021	60	1	0	125	258	0	0	141	1	2.8	1	1	3	0
1022	47	1	0	110	275	0	0	118	1	1.0	1	1	2	0
1023	50	0	0	110	254	0	0	159	0	0.0	2	0	2	1
1024	54	1	0	120	188	0	1	113	0	1.4	1	1	3	0

1025 rows × 14 columns

Dataset Input attributes

1. Age (in Years)
2. Sex (value 1: Male, value 0: Female)
3. cp: Chest Pain Type (1: typical angina, 2: atypical angina, 3: non-anginal pain, 4: asymptomatic)
4. Trest Blood Pressure (mm Hg on admission to the hospital)
5. Chol: cholesterol value
6. FBS: Fasting Blood Sugar (value 1:> 120 mg/dl; value 0:< 120 mg/dl)
7. Restecg: resting electrographic results (value 0: normal; value 1: 1 having ST-T wave abnormality, value 2: showing probable or definite left ventricular hypertrophy)
8. Thalach-maximum heart rate achieved
9. Exang-exercise-induced angina (value 1: yes, value 0: no)
10. Oldpeak-ST depression induced by exercise relative to rest
11. Slope the slope of the peak exercise ST segment (value 1: unsloping: value 2: flat; value 3 downsloping)
12. CA number of major vessels colored by fluoroscopy (value 0-3)
13. Thal (value 3: normal; value 6: fixed defect; value 7:reversible defect)
14. target: 1 for the presence of heart disease and 0 for the absence of heart disease

The data set is in CSV (Comma Separated Value) format which is further prepared to data frame as supported by the pandas library in python.

5.3 Data Preprocessing

We have used the dataset from the Kaggle website. The dataset is already pre-processed as it does not contain any null values or duplicates and it also does not have any outliers.

Hence, we are not doing any preprocessing of the dataset taken.

5.4 Feature Selection

In machine learning selecting important features in the data is an important part of the full cycle.

Passing data with irrelevant features might affect the performance of the model because the model learns the irrelevant features passed in it.

Need of Feature Selection:

It helps simplify models to make them easier and faster to train.

Reduces training times.

Enhanced generalization by reducing overfitting

There are three types of feature selection

1)Filter method

2)Wrapper method

3)Embedded Method

We are going to use Embedded methods for the feature selection.

Embedded Methods

1. In Embedded Methods, the feature selection algorithm is integrated as part of the learning algorithm.
2. Embedded methods combine the qualities of filter and wrapper methods.
3. It is implemented by algorithms that have their own feature selection methods in them.
4. A learning algorithm takes advantage of its own variable selection process and performs feature selection and classification/regression at the same time.
5. The most Common embedded techniques are the tree algorithms like Random Forest, Decision Tree, and so on.
6. Tree algorithms select a feature in each recursive step of the tree growth process and divide the sample set into smaller subsets. The more child nodes in a subset are in the same class, the more informative the features are.

Advantages of Embedded Methods:

- They take into consideration the interaction of features like Wrapper Methods do.
- They are faster than Filter Methods.
- They are more accurate than Filter Methods.
- They find the feature subset for the algorithm being trained.
- They are much less prone to over-fitting.

Algorithm-Based Approach:

- This can be done using any kind of tree-based algorithm like Decision Tree, Random Forest, and so on.
- The split takes place on a feature within the algorithm to find the correct variable.
- The algorithm tries all possible ways of splitting for all the features and chooses the one that splits the data best. This basically means it uses the wrapper method as all the possible combinations of features are tried and the best one is picked.
- With the help of this method, we can find feature importances and can remove features below a certain threshold.

5.5 Splitting the dataset

The splitting of a dataset refers to the process of dividing a dataset into two or more subsets. The most common split is into a training dataset and a testing dataset. The training dataset is used to train the machine learning model, while the testing dataset is used to evaluate the performance of the trained model.

```
In [11]: features=[ 'age','sex','cp','trestbps','chol','fbs','restecg','thalach','exang','oldpeak','slope','ca',
         X=ps.loc[:, features]
         y=ps.loc[:,['target']]
```

where ‘X’ represents the input data, which are the features or attributes of the data. e.g.: here the features are the patient's details such as age, sex, cp, etc.

And ‘y’ represents the output or target variable, which is the value that we want to predict.

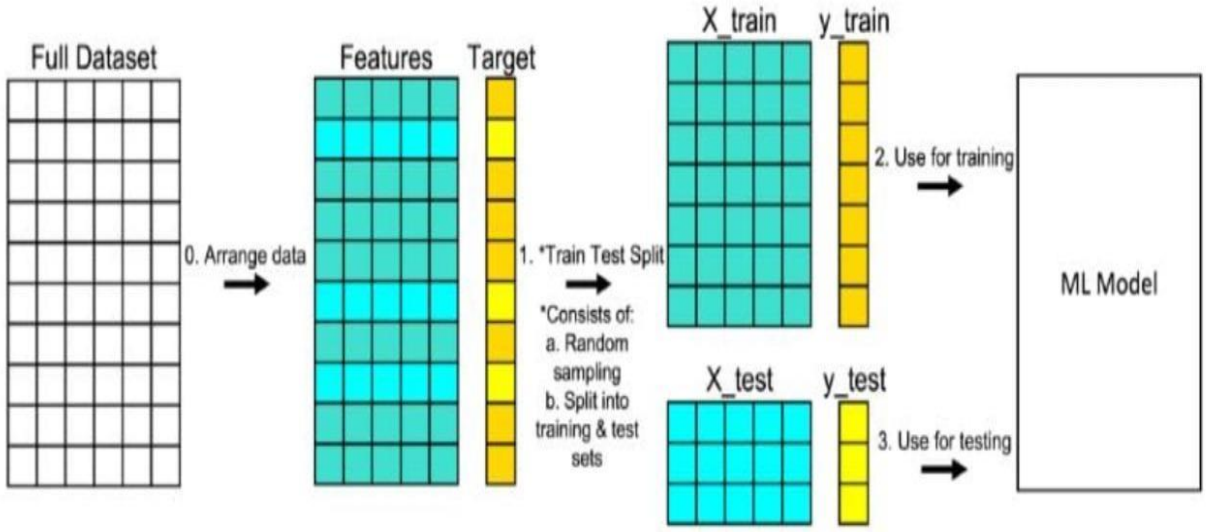
A training dataset is a set of data used to train a machine-learning model. The model learns from this data and uses it to make predictions on new, unseen data.

A testing dataset is a separate set of data used to evaluate the performance of the trained model. The model is applied to the testing dataset and the results are compared to the known outcomes to evaluate the model's accuracy.

The most common method of splitting a dataset is called random sampling, where each data point has an equal chance of being placed in either the training or testing dataset. The percentage of the dataset allocated to the training set and testing set is determined by the user, with a common ratio being 80/20, 75/25, or 60/40 where 75% is used for training and 25% is used for testing.

e.g.

```
In [19]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0, train_size = .75)
```



Now we are going to train the model using the training dataset and apply the following algorithms.

5.6 Algorithm Used

5.6.1 SUPPORT VECTOR MACHINE (SVM):

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed a Support Vector Machine.
.
The followings are important concepts in SVM -

Support Vectors - Data Points that are closest to the hyperplane are called support vectors. Separating lines will be defined with the help of these data points.

Hyperplane - As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

Margin - It may be defined as the gap between two lines on the closest data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. A large margin is considered a good margin and a small margin is considered a bad margin.

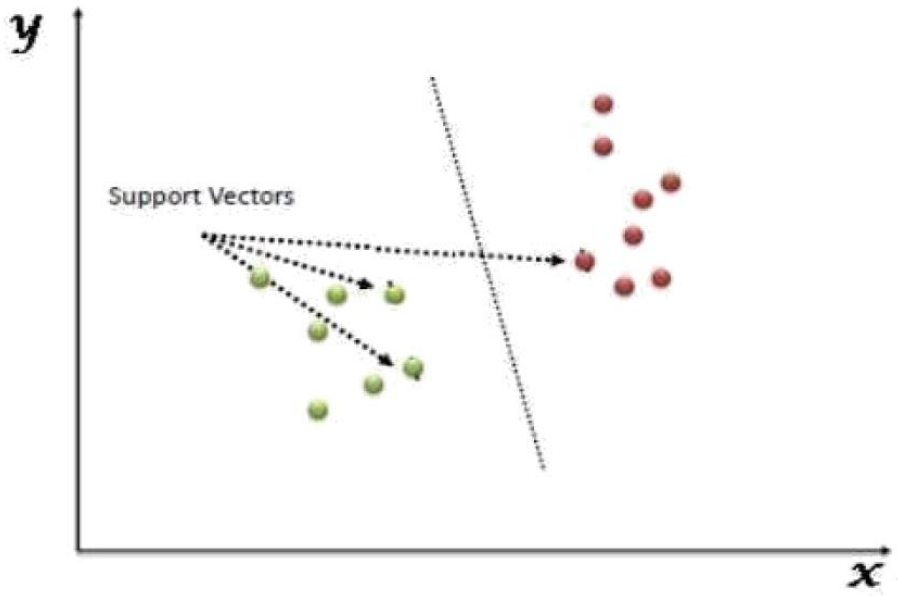
Types of SVM:

SVM can be of two types:

● Linear SVM:
Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and the classifier is used as Linear SVM classifier.

● Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data, and the classifier used is called a Non-linear SVM classifier.

The objective of the support vector machine algorithm is to find a hyperplane in N-dimensional space (N - the number of features) that distinctly classifies the data points.



The advantages of support vector machines are :

● Effective in high dimensional spaces.
● Still effective in cases where the number of dimensions is greater than the number of samples.
● Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
● Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

5.6.2 NAIVE BAYES ALGORITHM:

Naive Bayes algorithm is a supervised learning algorithm, which is based on the Bayes theorem and used for solving classification problems. It is mainly used in text classification that includes a high-dimensional training dataset.
Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.
The Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.
The Naive Bayes algorithm is comprised of two words Naive and Bayes, Which can be described as

- **Naive**: It is called Naive because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the basis of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identifying that it is an apple without depending on each other.
- **Bayes:** It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes's theorem:
Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
The formula for Bayes' theorem is given as

$$P(A|B) = P(B|A)\, P(A)\, /P(B)$$

Where,
P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.
P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.
P(A) is Prior Probability: The probability of the hypothesis before observing the evidence.
P(B) is Marginal Probability: Probability of Evidence.

Types of Naive Bayes models:

There are three types of Naive Bayes Model, which are given below:
• **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors take continuous values instead of discrete ones, then the model assumes that these values are sampled from the Gaussian distribution.

• **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for document classification problems, it means a particular document belongs to which category such as Sports, Politics, education, etc. The classifier uses the frequency of words for the predictors.

• **Bernoulli**: The Bernoulli classifier works similarly to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

5.7 Model Accuracy and Testing

Now as we have trained our model using machine learning algorithms, we are going to test our model using the test dataset which we split earlier from the original dataset. Now the model will predict the output which is whether the patient has heart disease or not.

Compare the predictions to the actual labels (i.e., whether the patient has heart disease or not) in the test set to evaluate the model's performance. Common

metrics for classification problems include confusion, accuracy, precision, recall, and F1 score.

Now let us calculate the accuracy of our trained model There are several ways to calculate the accuracy of a trained machine learning model for predicting heart disease, but one of the most common methods is to use the following formula:

Accuracy = (number of correct predictions) / (total number of predictions)

To calculate accuracy, you can compare the model's predictions to the actual labels (i.e., whether the patient has heart disease or not) in the test set.

For example,

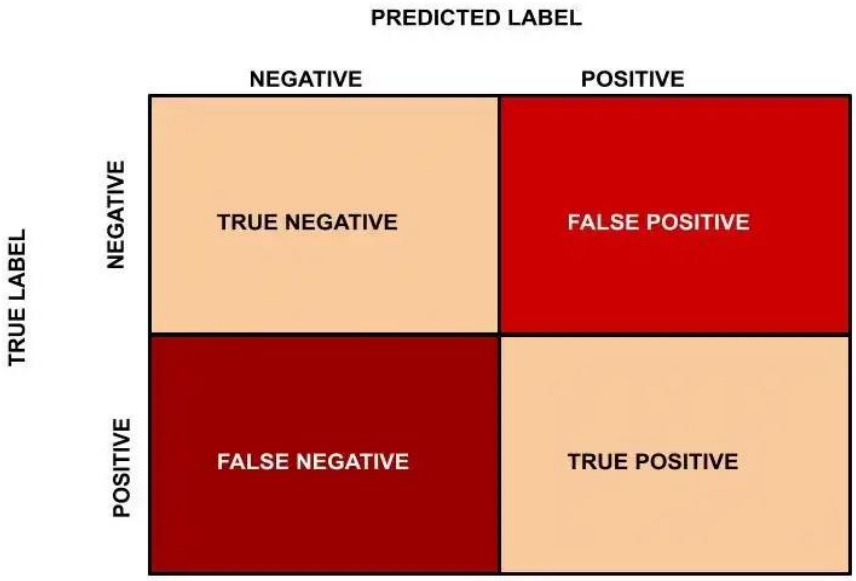
- if the model correctly predicts that a patient has heart disease and the patient has heart disease, that is considered a **true positive**.

- If the model correctly predicts that a patient does not have heart disease and the patient does not have heart disease, that is considered a **true negative**.

- If the model incorrectly predicts that a patient has heart disease and the patient does not have heart disease, that is considered a **false positive.**

- If the model incorrectly predicts that a patient does not have heart disease and the patient has heart disease, that is considered a **false negative.**

Now we will introduce the ***confusion matrix*** which is required to compute the ***accuracy*** of the machine learning algorithm in classifying the data into its corresponding labels.



The formula for accuracy can be expressed in terms of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) as follows:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

We are going to use libraries such as scikit-learn, which have inbuilt functions to calculate accuracy, precision, recall, and f1-score.

It's important to note that accuracy is not the only metric that should be considered when evaluating a machine-learning model. Other important metrics include precision, recall, and F1 score. These metrics provide a more complete picture of the model's performance, especially when dealing with imbalanced datasets.

Precision-:

Precision is a metric used to evaluate the performance of a machine learning model, specifically in classification problems. It is a measure of the number of true positive predictions made by the model out of the total number of positive predictions made by the model.

Formally, precision is defined as

Precision = True Positives / (True Positives + False Positives)

where:

True Positives (TP) are the number of times the model correctly predicts that a patient has heart disease.

False Positives (FP) are the number of times the model incorrectly predicts that a patient has heart disease when the patient actually does not have heart disease.

In other words, precision tells us how often the model is correct when it predicts that a patient has heart disease. High precision means that the model has a low rate of false positives, which is desirable in many cases, such as medical diagnosis.

Recall-:

Recall is also a metric used to evaluate the performance of a machine learning model. It is a measure of the number of true positive predictions made by the model out of the total number of actual positive cases.

Formally, recall is defined as

Recall = True Positives / (True Positives + False Negatives)

where:

False Negatives (FN) are the number of times the model incorrectly predicts that a patient does not have heart disease when the patient actually has heart disease.

In other words, recall tells us how many of the actual positive cases the model can identify. High recall means that the model has a low rate of false negatives, which is desirable in many cases, such as medical diagnoses.

F1 score-:

The F1 score is also a metric used to evaluate the performance of a machine learning model. It is a measure that combines precision and recall, and it is calculated as the harmonic mean of precision and recall.

Formally, the F1 score is defined as

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

The F1 score ranges from 0 to 1, where a score of 1 indicates perfect precision and recall, and a score of 0 indicates that the model has not made any true positive predictions.

6. Tools and Techniques

Algorithms used: Support Vector Machine, Naive Bayes.

Libraries Used: Pandas, NumPy, skLearn, matplotlib.

Integrated Development Environment (IDE): Jupyter Notebook.

7. Expected Outcome of the Research

The expected outcome for a heart disease prediction system is to accurately identify individuals who are at high risk for developing heart disease, based on various risk factors such as age, gender, family history, cholesterol levels, and blood pressure. This information can then be used by healthcare providers to make informed decisions about preventative measures and treatment options for those individuals. The goal is to reduce the incidence of heart disease and improve patient outcomes.

At last, we will compare the accuracy result of both algorithms and will determine the algorithm which gives better results.

8. References

https://ieeexplore.ieee.org/document/9734880

https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012046

https://towardsdatascience.com/predicting-presence-of-heart-diseases-using-machine-learning-36f00f3edb2c

https://medium.com/analytics-vidhya/feature-selection-for-dimensionality-reduction-embedded-method-e05c74014aa

https://www.researchgate.net

Bashir S, Qamar U & Javed M Y (2014, November). An ensemble-based decision support framework for intelligent heart disease diagnosis. In International Conference on Information Society (i-Society 2014) (pp. 259-64). IEEE.

https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd

https://builtin.com/data-science/train-test-split